php

Search

simplexml_load_file

Change language: French

Edit Report a Bug

(PHP 5)
simplexml_load_file — Convertit un fichier XML en objet

Description

```
SimpleXMLElement simplexml_load_file ( string $filename [, string $class_name = "SimpleXMLElement" [, int $options = 0 [, string $ns = "" [, bool $is_prefix = false ]]]] )
```

Convertit le document XML **filename** en un objet de type SimpleXMLElement.

Liste de paramètres

filename

Chemin vers le fichier XML

> **Note**:
> Libxml 2 supprime la protection des caractères des URI, alors si vous voulez passer par exemple *b&c* comme paramètre URI à *a*, vous devez appeler *simplexml_load_file(rawurlencode('http://example.com/?a='.urlencode('b&c')))*. Depuis PHP 5.1.0, vous n'avez plus besoin de faire cela puisque PHP le fait pour vous.

class_name

Vous pouvez utiliser ce paramètre optionnel et ainsi, la fonction **simplexml_load_file()** retournera un objet de la classe spécifiée. Cette classe doit étendre la classe SimpleXMLElement.

options

Depuis PHP 5.1.0 et Libxml 2.6.0, vous pouvez aussi utiliser le paramètre **options** pour spécifier des paramètres additionnels Libxml.

ns

Préfixe ou l'URI de l'espace de noms.

is_prefix

TRUE si **ns** est un préfixe, FALSE si c'est l'URI ; par défaut, FALSE.

Valeurs de retour

Retourne un objet de la classe SimpleXMLElement dont les propriétés contiennent les données du document XML, ou FALSE si une erreur survient.

Erreurs / Exceptions

Produit un message d'erreur de niveau E_WARNING pour chaque erreur trouvée dans les données XML.

Astuce Utilisez la fonction libxml_use_internal_errors() pour supprimer toutes les erreurs XML, et la fonction libxml_get_errors() pour les parcourir.

Exemples

Exemple #1 Interprétation d'un document XML

```
<?php
// Le fichier test.xml contient un document XML avec un élément racine
// et au moins un élément /[racine]/title.

if (file_exists('test.xml')) {
    $xml = simplexml_load_file('test.xml');

    print_r($xml);
} else {
    exit('Echec lors de l\'ouverture du fichier test.xml.');
}
?>
```

Ce script affichera, en cas de succès :

```
SimpleXMLElement Object
(
  [title] => Example Title
  ...
)
```

À partir de là, vous pouvez utiliser *$xml->title* et tout autre élément.

Voir aussi

- simplexml_load_string() - Convertit une chaîne XML en objet
- SimpleXMLElement::__construct() - Crée un nouvel objet SimpleXMLElement
- Traitement des erreurs XML
- libxml_use_internal_errors() - Désactive le rapport d'erreur libxml et les stocke pour lecture ultérieure
- Utilisation de base SimpleXML

User Contributed Notes 24 notes

add a note

▲ 5 ▼ fdouteaud at gmail dot com 8 years ago

Be careful if you are using simplexml data directly to feed your MySQL database using MYSQLi and bind parameters.

The data coming from simplexml are Objects and the bind parameters functions of MySQLi do NOT like that! (it causes some memory leak and can crash Apache/PHP)

In order to do this properly you MUST cast your values to the right type (string, integer...) before passing them to the binding methods of MySQLi.
I did not find that in the documentation and it caused me a lot of headache.

▲ 8 ▼ wouter at code-b dot nl 7 years ago

To correctly extract a value from a CDATA just make sure you cast the SimpleXML Element to a string value by using the cast operator:

```
<?php
$xml = '<?xml version="1.0" encoding="UTF-8" ?>
<rss>
  <channel>
    <item>
      <title><![CDATA[Tom & Jerry]]></title>
    </item>
  </channel>
</rss>';

$xml = simplexml_load_string($xml);

// echo does the casting for you
echo $xml->channel->item->title;

// but vardump (or print_r) not!
var_dump($xml->channel->item->title);

// so cast the SimpleXML Element to 'string' solve this issue
var_dump((string) $xml->channel->item->title);
?>
```

Above will output:

Tom & Jerry

object(SimpleXMLElement)#4 (0) {}

string(11) "Tom & Jerry"

▲ 3 ▼ sean at aliencreations dot com 3 years ago

If you find that you are receiving 500 errors with simplexml_load_file() but you can access the xml/rss feed manually through a browser, your script is probably being blocked by a user agent sniffer.

Add this code before your xml call to remedy this issue

```
<?php

ini_set("user_agent","Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)");
ini_set("max_execution_time", 0);
ini_set("memory_limit", "10000M");

$rss = simplexml_load_file($feed_url);

?>
```

▲ 2 ▼ Kyle 7 years ago

In regards to Anonymous on 7th April 2006

There is a way to get back HTML tags. For example:

```
<?xml version="1.0"?>
<intro>
   Welcome to <b>Example.com</b>!
</intro>

<?php
// I use @ so that it doesn't spit out content of my XML in an error message if the load fails. The content could be passwords so this is just to be safe.
$xml = @simplexml_load_file('content_intro.xml');
if ($xml) {
   // asXML() will keep the HTML tags but it will also keep the parent tag <intro> so I strip them out with a str_replace. You could obviously also use a preg_replace if you have lots of tags.
   $intro = str_replace(array('<intro>', '</intro>'), '', $xml->asXML());
} else {
   $error = "Could not load intro XML file.";
}
?>
```

With this method someone can change the intro in content_intro.xml and ensure that the HTML is well formed and not ruin the whole site design.

▲ 4 ▼ ricardo at ricardomartins dot info 1 year ago

Sometimes we have xml's with hyphens nodes, like

```
<my_xml>
 <some-node>value</some-node>
</my_xml>
```

You'll need to use

```
<?php
$simpleXmlObj->{'some-node'}
?>
```

instead of

```
<?php
$simpleXmlObj->some-node;
?>
```

▲ 1 ▼ l [DOT] anzinger [AT] gmail [DOT] com 6 years ago

If you don't want that the CDATA values get escaped, just load the XML with LIBXML_NOCDATA as an 3rd argument.

Note: A PHP version >= 5.1.0 is required for this to work.

Example:

```
<?php simplexml_load_file('xmldatei.xml', null, LIBXML_NOCDATA); ?>
```

▲ 1 ▼ skutter at imprecision dot net 8 years ago

So it seems SimpleXML doesn't support CDATA... I bashed together this little regex function to sort out the CDATA before trying to parse XML with the likes of simplexml_load_file / simplexml_load_string. Hope it might help somebody and would be very interested to hear of better solutions. (Other than *not* using SimpleXML of course! ;)

It looks for any <![CDATA [Text and HTML etc in here]]> elements, htmlspecialchar()'s the encapsulated data and then strips the "<![CDATA [" and "]]>" tags out.

```
<?php
function simplexml_unCDATAise($xml) {
   $new_xml = NULL;
   preg_match_all("/\<\!\[CDATA \[(.*)\]\]\>/U", $xml, $args);

   if (is_array($args)) {
      if (isset($args[0]) && isset($args[1])) {
         $new_xml = $xml;
         for ($i=0; $i<count($args[0]); $i++) {
            $old_text = $args[0][$i];
            $new_text = htmlspecialchars($args[1][$i]);
            $new_xml = str_replace($old_text, $new_text, $new_xml);
         }
      }
   }

   return $new_xml;
}

//Usage:
$xml = 'Your XML with CDATA...';
$xml = simplexml_unCDATAise($xml);
$xml_object = simplexml_load_string($xml);
?>
```

▲ 1 ▼ neil art neilanddeb dort com 4 years ago

Because the encoding of my XML file is UTF-8 and the
encoding of my web page is iso-8859-1 I was getting strange characters such as â€™ instead of a right single quote.

The solution to this turned out to be hard to find, but really easy to implement.

http://uk3.php.net/manual/en/function.iconv.php

Using the iconv() function you can convert from one encodign to another, the TRANSLIT option seems to work best for what I needed.
Here's my example:

```
<?php
// convert string from utf-8 to iso8859-1
$horoscope = iconv( "UTF-8", "ISO-8859-1//TRANSLIT", $horoscope );
?>
```

I found the solution on this page...
http://tinyurl.com/lm39xc
Hope this helps

▲ 0 ▼ kannan at 99deals dot in 1 year ago

If you have some nodes which are having special characters, it would not load properly

for an instance see the nodes below
<node:number>1538-7445</node:number>
<node:coverDisplayDate>Sep 1 2012 12:00:00:000AM</node:coverDisplayDate>

either you have to change the : to other special characters like '-' in order to convert it properly

Correct Node
<node-number>1538-7445</node-number>
<node-coverDisplayDate>Sep 1 2012 12:00:00:000AM</node-coverDisplayDate>

I have wasted my precious time while debugging this. Please aware about this. ?

▲ 0 ▼ Nanuit at ossi dot at 3 years ago

A little function very helpfull in using simplexml_load_file behind a proxy

```
<?php
function getXMLfromURL($url) {
    $Proxy = getenv("HTTP_PROXY");

    if (strlen($Proxy) > 1) {
     $r_default_context = stream_context_get_default ( array
              ('http' => array(
                 'proxy' => $Proxy,
                 'request_fulluri' => True,
              ),
           )
        );
     libxml_set_streams_context($r_default_context);
    }
    $daten = simplexml_load_file($url);
    return ($daten);
  }
?>
```

where HTTP_PROXY is set to e.g.: tcp://proxy:8080

▲ 0 ▼ raduispas at gmail dot com 3 years ago

if you want to check when this function fails,make sure to compare the return value with === instead of == :

```
<?php
$url = 'http://www.example.com';
$xml = simpleXML_load_file($url,"SimpleXMLElement",LIBXML_NOCDATA);
if($xml ===  FALSE)
{
  //deal with error
}
else { //do stuff }
?>
```

Otherwise you may end up with FALSE all the time even if the document is ok. Hope this helps someone ;)

▲ 0 ▼ jamie at splooshmedia dot co dot uk 4 years ago

A wrapper around simplexml_load_file to circumvent nasty error messages when the xml server times out or gives a 500 error etc.

```
<?php
function loadXML2($domain, $path, $timeout = 30) {

  /*
    Usage:
```

```
    $xml = loadXML2("127.0.0.1", "/path/to/xml/server.php?code=do_something");
    if($xml) {
       // xml doc loaded
    } else {
       // failed. show friendly error message.
    }
*/

$fp = fsockopen($domain, 80, $errno, $errstr, $timeout);
if($fp) {
    // make request
    $out = "GET $path HTTP/1.1\r\n";
    $out .= "Host: $domain\r\n";
    $out .= "Connection: Close\r\n\r\n";
    fwrite($fp, $out);

    // get response
    $resp = "";
    while (!feof($fp)) {
       $resp .= fgets($fp, 128);
    }
    fclose($fp);
    // check status is 200
    $status_regex = "/HTTP\/1\.\d\s(\d+)/";
    if(preg_match($status_regex, $resp, $matches) && $matches[1] == 200) {
       // load xml as object
       $parts = explode("\r\n\r\n", $resp);
       return simplexml_load_string($parts[1]);
    }
}
return false;

}
?>
```

▲ 0 ▼ Smokey 4 years ago

for nested and same name values i'v made up this little bit for getting and displaying multiable values from google's geocode when a exact match is not found it returns all close matches in the following format(this is an abriged version of there output)

```
<Response>
 <Placemark id="1">
  <address> New York 24, NY, USA</address>
  <AddressDetails>
   ..................
  </AddressDetails>
  <Point>
   <coordinates>-73.5850086,40.7207442,0</coordinates>
  </Point>
 </Placemark>
 <Placemark id="2">
  <address>New York 27, NY, USA</address>
  <AddressDetails>
   ..................
  </AddressDetails>
  <Point>
   <coordinates>-72.8987835,40.8003588,0</coordinates>
  </Point>
```

```
  </Placemark>
  <Placemark id="3">
    <address>Cedar Place School, 20 Cedar Pl, Yonkers, NY 10705, USA</address>
    <AddressDetails>
      ..................
    </AddressDetails>
    <Point>
      <coordinates>-73.8966320,40.9256520,0</coordinates>
    </Point>
  </Placemark>
</Response>

<?php
// get and breakdown the results then store them in $var's
$Address = "99999 parkplace, new york, NY";
$urladdress = urlencode($Address);
$Base_url = "http://maps.google.com/maps/geo?q=";
$urlParts = "&output=xml";
$urlrequest = $Base_url . $urladdress . $urlParts;
$xml = simplexml_load_file($urlrequest);
$num = "0";
foreach ($xml->Response->Placemark as $value){
   $num++;
   $GeoFindAdd{$num} = $value->address;
   $GeoFindCords{$num} = $value->Point->coordinates;
}

// a simple display for the results
echo "Found ",$num," Possable Geo Data Sets <br>";
$CountNumResults = "0";
for ( ; $num > 0; $num--){
   $CountNumResults++;
   echo $countnum,"<br> Address = ",$GeoFindAdd{$num},"<br> Coordinates = ",$GeoFindCords{$num},"<br>";
}
echo "END";
?>
```

▲ 0 ▼ knl at bitflop dot com 4 years ago

If you need to parse the data from SimpleXML into a session variable remember to define the data as a string first.

If you don't you will get warnings of "Node no longer exists" pointing to your session_start() function.

This will work:

```
<?php

   $new_version = simplexml_load_file('http://example.com/version.xml');
   $_SESSION['current_version'] = (string)$new_version->version;

?>
```

▲ 0 ▼ cryonyx at cerebrate dot ru 5 years ago

In case you have a XML file with a series of equally named elements on one level simplexml incorrectly processes them and doesn't allow to walk through the array using foreach(). As far as I'm concerned, it is the problem caused by PHP xml_parser (see: http://ru2.php.net/manual/ru/function.xml-parser-create.php#53188).

To avoid this, just use count() and walk through the array using for().

Example:

```
<params>
 <param>
  <name>version.shell</name>
  <value>1.0</value>
 </param>
 <param>
    <name>version.core</name>
    <value>1.0</value>
 </param>
 <param>
    <name>file.lang</name>
    <value>vc.lang</value>
 </param>
 ...
</params>

<?php
$filename = '...';
$xml = simplexml_load_file($filename);
$p_cnt = count($xml->param);
for($i = 0; $i < $p_cnt; $i++) {
 $param = $xml->param[$i];
 ...;
}
?>
```

▲ 0 ▼ mario 5 years ago

If you want CDATA in your object you should use LIBXML_NOCDATA

```
<?php
$xml = simplexml_load_file($file_xml, 'SimpleXMLElement',LIBXML_NOCDATA);

   print_r($xml);
?>
```

▲ 0 ▼ php at werner dash ott dot de 7 years ago

Making SimpleXMLElement objects session save.

Besides the effect of not surviving sessions, the SimpleXMLElement object may even crash the session_start() function when trying to re-enter the session!

To come up with a solution for this, I used a pattern as follows. The core idea is to transform the SimpleXMLElement between session calls to and from a string representation which of course is session save.

```
<?php
 //
 // session save handling of SimpleXMLElement objects
 // (applies to/ tested with PHP 5.1.5 and PHP 5.2.1)
 // The myClass pattern allows for conveniently accessing
 // XML structures while being session save
 //
 class myClass
```

```
 {
  private $o_XMLconfig = null;
  private $s_XMLconfig = '';

  public function __construct($args_configfile)
  {
   $this->o_XMLconfig = simplexml_load_file($args_configfile);
   $this->s_XMLconfig = $this->o_XMLconfig->asXML();
  } // __construct()

  public function __destruct()
  {
   $this->s_XMLconfig = $this->o_XMLconfig->asXML();
   unset($this->o_XMLconfig); // this object would otherwise crash
                              // the subsequent call of
                              // session_start()!
  } // __destruct()

  public function __wakeup()
  {
   $this->o_XMLconfig = simplexml_load_string($this->s_XMLconfig);
  } // __wakeup()

 } // class myClass
?>
```

▲ 0 ▼ Anonymous 8 years ago

What has been found when using the script is that simplexml_load_file() will remove any HTML formating inside the XML file, and will also only load so many layers deep. If your XML file is to deap, it will return a boolean false.

▲ 0 ▼ info at evasion dot cc 8 years ago

Sorry there's a mistake in the previous function :

```
<?php
 function &getXMLnode($object, $param) {
    foreach($object as $key => $value) {
       if(isset($object->$key->$param)) {
          return $object->$key->$param;
       }
       if(is_object($object->$key)&&!empty($object->$key)) {
          $new_obj = $object->$key;
          // Must use getXMLnode function there (recursive)
          $ret = getXMLnode($new_obj, $param);

       }
    }
    if($ret) return (string) $ret;
    return false;
 }
?>
```

▲ 0 ▼ info at evasion dot cc 8 years ago

Suppose you have loaded a XML file into $simpleXML_obj.
The structure is like below :

SimpleXMLElement Object

```
(

   [node1] => SimpleXMLElement Object
      (
         [subnode1] => value1
         [subnode2] => value2
         [subnode3] => value3
      )

   [node2] => SimpleXMLElement Object
      (
         [subnode4] => value4
         [subnode5] => value5
         [subnode6] => value6
      )

)
```

When searching a specific node in the object, you may use this function :

```
<?php

   function &getXMLnode($object, $param) {
      foreach($object as $key => $value) {
         if(isset($object->$key->$param)) {
            return $object->$key->$param;
         }
         if(is_object($object->$key)&&!empty($object->$key)) {
            $new_obj = $object->$key;
            $ret = getCfgParam($new_obj, $param);
         }
      }
      if($ret) return (string) $ret;
      return false;
   }
?>
```

So if you want to get subnode4 value you may use this function like this :

```
<?php
$result = getXMLnode($simpleXML_obj, 'subnode4');
echo $result;
?>
```

It display "value4"

▲ -1 ▼ genialbrainmachine at NOSPAM dot tiscali dot it 8 years ago

Micro$oft Word uses non-standard characters and they create problems in using simplexml_load_file.
Many systems include non-standard Word character in their implementation of ISO-8859-1. So an XML document containing that characters can appear well-formed (i.e.) to many browsers. But if you try to load this kind of documents with simplexml_load_file you'll have a little bunch of troubles..
I believe that this is exactly the same question discussed in htmlentites. Following notes to htmlentitles are interesting here too (given in the reverse order, to grant the history):
http://it.php.net/manual/en/function.htmlentities.php#26379
http://it.php.net/manual/en/function.htmlentities.php#41152
http://it.php.net/manual/en/function.htmlentities.php#42126
http://it.php.net/manual/en/function.htmlentities.php#42511

-1 fusionstream at gmail dot com 2 years ago

If you are loading many files, this may slow down your page load time.

To set a timeout, use file_get_context and then simplexml_load_string

```
<?php

$fp = fopen('http://www.example.com/rss', false, stream_create_context(array('http' => array('timeout', '1.5'))));

if ($fp) {
   print_r( simplexml_load_string($fp) );
} else {
   echo "The request timed out";
}
?>
```

-1 mark 8 years ago

If the property of an object is empty the array is not created. Here is a version object2array that transfers properly.

```
<?php
function object2array($object)
{
   $return = NULL;

   if(is_array($object))
   {
      foreach($object as $key => $value)
         $return[$key] = object2array($value);
   }
   else
   {
      $var = get_object_vars($object);

      if($var)
      {
         foreach($var as $key => $value)
            $return[$key] = ($key && !$value) ? NULL : object2array($value);
      }
      else return $object;
   }

   return $return;
}
?>
```

-6 guego dot ol at ig dot com dot br 3 years ago

Analyze fully XML.

```
<?php
$xml = simplexml_load_file('file.xml');

foreach($xml as $key0 => $value){
echo "..1..[$key0] => $value";
foreach($value->attributes() as $attributeskey0 => $attributesvalue1){
```

```
echo "_________[$attributeskey0] = $attributesvalue1";
}
echo '<br />';
////////////////////////////////////////////////
foreach($value as $key => $value2){
echo "....2.....[$key] => $value2";
foreach($value2->attributes() as $attributeskey => $attributesvalue2){
echo "_________[$attributeskey] = $attributesvalue2";
}
echo '<br />';
////////////////////////////////////////////////
foreach($value2 as $key2 => $value3){
echo ".........3.........[$key2] => $value3";
foreach($value3->attributes() as $attributeskey2 => $attributesvalue3){
echo "_________[$attributeskey2] = $attributesvalue3";
}
echo '<br />';
////////////////////////////////////////////////
foreach($value3 as $key3 => $value4){
echo "...................4.....................[$key3] => $value4";
foreach($value4->attributes() as $attributeskey3 => $attributesvalue4){
echo "_________[$attributeskey3] = $attributesvalue4";
}
echo '<br />';
////////////////////////////////////////////////
foreach($value4 as $key4 => $value5){
echo ".....................5.....................[$key4] => $value5";
foreach($value5->attributes() as $attributeskey4 => $attributesvalue5){
echo "_________[$attributeskey4] = $attributesvalue5";
}
echo '<br />';
////////////////////////////////////////////////
foreach($value5 as $key5 => $value6){
echo ".....................6.....................[$key5] => $value6";
foreach($value6->attributes() as $attributeskey5 => $attributesvalue6){
echo "_________[$attributeskey5] = $attributesvalue6";
}
echo '<br />';
}}}}}
echo '<br />';
}
?>
```

add a note

Fonctions SimpleXML

simplexml_import_dom
» **simplexml_load_file**
simplexml_load_string

Copyright © 2001-2014 The PHP Group My PHP.net Contact Other PHP.net sites Mirror sites

Privacy policy